Network Working Group Brian Bidulock
INTERNET-DRAFT OpenSS7 Corporation

Expires in six months January 2, 2003

Load Selection (LOADSEL)
for
Signalling User Adaptation Layers
<draft-bidulock-sigtran-loadsel-01.ps>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 or RFC 2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as 'work in progress'.

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

To learn the current status of any Internet-Draft, please check the Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

This Internet-Draft describes *Load Selection* (*LOADSEL*) for Signalling User Adaptation Protocols [IUA...TUA], which permits an Application Server Processes (ASP) to indicate its placement within an Application Server and permits an Signalling Gateway (SG) to distribute traffic over ASPs in Application Servers under Application Server Process (ASP) control.

1. Introduction

1.1. Scope

This Internet-Draft provides parameters and associated procedures in extension to the parameters and procedures of the Signalling User Adaptation Layers (UAs) [IUA...TUA], for the purpose of permitting Application Server Process control over placement of ASPs within Application Servers (or Load Groups) [LOADGRP] as part of the normal procedures of these UA protocols.

UA implementations supporting *LOADSEL* are intended to be compatible with UA implementations not supporting *LOADSEL*.

1.2. Change History

1.2.1. Changes from Version 0.0 to Version 1.0

- added this section,
- updated release version and dates,
- updated references,
- updated postscript diagrams,
- minor formatting changes,
- updated author's address.

1.3. Terminology

LOADSEL supplements the terminology used in the UA documents [IUA...TUA] by adding the following terms:

Load Selection – a partition of traffic within an Application Server described by the *Load Selection* parameter and identified by a *Load Selector* parameter.

Load Selector (LS) – an identifier that uniquely identifies a partition of traffic flow within an Application Server. This identifier is only guaranteed unique within the scope of an Application Server and must be combined with a *Routing Context* (or *Interface Identifier(s)*) (explicitly or implicitly) to uniquely define a partition of traffic flow at a Signalling Gateway.

Signalling User Adaptation Layer (UA) – one or more of the Stream Control Transmission Protocol (SCTP) [RFC 2960] ISDN Signalling User Adaptation Layers [IUA...GR303UA] or SS7 Signalling User Adaptation Layers [M2UA...TUA] supporting the concept of an Application Server.

1.4. Overview

UA procedures with regard to traffic distribution and ASP traffic management provide a mechanism to select the algorithm for coordinating state and distributing traffic over a number of Application Server Processes (ASPs) serving an Application Server (AS). These existing procedures provide only simplified distribution approaches which are not amenable in the following situations:

(1) Connection- or Transaction-Oriented traffic flows which group the messages corresponding to a connection or a transaction.

(2) Large scale systems that need to adapt to dynamic traffic loading.

(3) Systems that need dynamic reconfiguration under ASP control for maintenance or fail-over purposes.

(4) Systems that require ASP control over load-sharing.

LOADSEL for the Signalling User Adaptation Layers [IUA...TUA] permits close control over the placement and grouping of Application Server Processes serving an Application Server that provides for load selection placement within the Application Server: a capability not present in the existing scheme.

Under existing UA traffic distribution, there is no mechanism which permits an Application Server Process (ASP) to indicate which of a number of ASPs it wishes to override in an Override AS. There is also no mechanism which permits an ASP to indicate which traffic flows it wishes to process within a Load-share or Broadcast AS.

LOADSEL provides the ability for the ASP itself to control its placement within an AS, be informed of the failure of ASPs serving other load selections within the AS, and chose to activate itself to receive additional load selections within the AS. *LOADSEL* also permits grouping of ASPs within a load selection.

1.4.1. Non-Load Selection Traffic Distribution

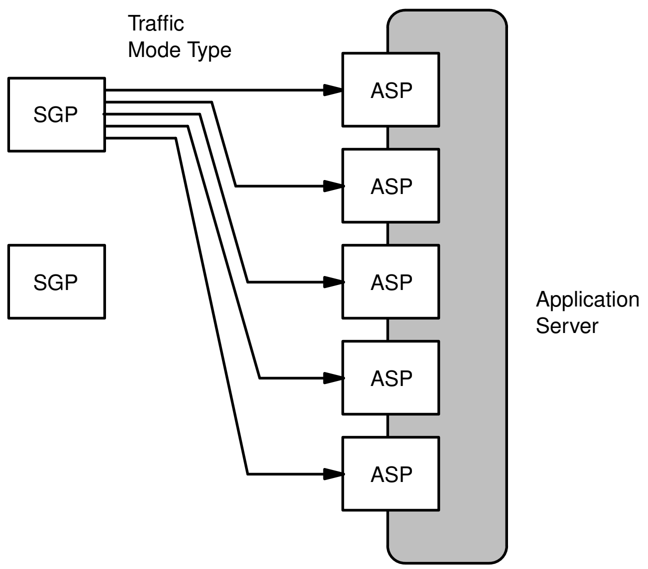

Figure 1. Non-Load Selection Traffic Distribution

Figure 1 illustrates the existing traffic distribution algorithm that is used across the Signalling User Adaptation Layers.

When an SGP distributes a Signalling User Adaptation Layer message toward the Application Server based on the *Routing (Link) Key*, it selects an ASP that is active for the AS according to the *Traffic Mode Type* that is associated with the AS. The *Traffic Mode Type* describes three general distribution algorithms: Override, Load-share and Broadcast.

The detailed actions taken for these distribution algorithms are described in Section 4 of the Signalling User Adaptation Layer specifications [IUA...TUA]; however, they can be summarized as follows:

Override:– When distributing messages to an Override Application Server, the SGP selects the ASP which is active for the Application Server. In an Override Application Server, at most one ASP can be active for the AS at any given point in time. The active ASP for the AS is selected.

A major limitation of the Override AS that is removed by *LOADSEL* is that only one ASP can be active within an Application Server. This does not work when the number of ASPs required to service an AS is greater than one.

Load-share:– When distributing messages to a Load-share Application Server, the SGP selects one of the ASPs that are active for the Application Server using an implementation dependent load-sharing algorithm based on some unspecified aspect of the traffic or static configuration data.

A major limitation of the Load-share AS is that each ASP in the Application Server must be capable of handling any (and all) traffic within the Load-share AS. Under the current approach an SGP does not distinguish between ASPs in a Load-share AS and, aside from perhaps attempting to equally distribute traffic load over the available ASPs, it is not possible for the ASPs to control which traffic flows it receives. This causes management difficulties when the number of ASPs required to service an AS is large, but the number of spare ASPs is small. It also causes considerable difficulty when each ASP does not have access to the entire AS call or transaction state.

Broadcast:– When distributing messages to a Broadcast Application Server, the SGP sends a copy of the message to each of the ASPs that are active for the Application Server. The ASPs themselves decide which, if any, of the ASPs will process the message.

A major limitation of the Broadcast AS is that each ASP receives each message. This does not scale well when the number of ASPs needed to service an AS is large.

LOADSEL enhances the traffic distribution algorithms of the existing Signalling User Adaptation Layers by providing ASP control over its placement within an AS by load selection.

1.4.2. Load Selection Traffic Distribution

Figure 2 illustrates the *LOADSEL* traffic distribution algorithms that are used across the Signalling User Adaptation Layers as a result of the *LOADSEL* messages and procedures.

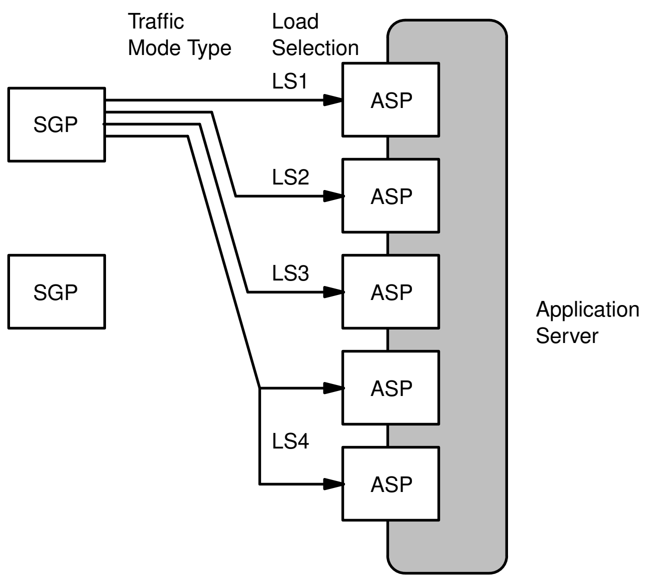

Figure 2. Load Selection Traffic Distribution

LOADSEL introduces the concept of a *Load Selector*. A *Load Selector* is an identifier that is used to identify a traffic flow within (or across) Application Server(s). Signalling Gateway Processes (SGPs) distribute traffic first over the *Load Selector* and then distribute traffic within the *Load Selector*. Each *Load Selector* describes (and is identified by) an identifier within the Application Server. The *Load Selector* identifies the traffic flows that will be distributed to ASPs associated with the *Load Selector* within an Application Server.

When an SGP distributes a Signalling User Adaptation Layer message toward an Application Server based on the *Routing (Link) Key*, it first derives a *Load Selector* according to a UA-specific, AS-specific and implementation-dependent load selection algorithm. This load selection algorithm is configured at the SGP and may consist, for example, of the Signalling Link Selection (SLS) [Q.704], Circuit Identification Code (CIC) [Q.723, Q.763], Subsystem Number (SSN) [Q.713], Destination Local Reference (DLR) [Q.713], or Transaction Id (TID) [Q.773], value contained in the message for distribution.

Once the SGP has determined the *Load Selector* to which the message corresponds, an ASP that is active for the associatd Load Selection within the AS is selected. When multiple ASPs are active for the same Load Selection within an AS, the SGP uses the traffic handling mode based on the *Traffic Mode Type* associated with the Application Server (or Load Group[1]) to choose the ASP within the load selection.

Under *LOADSEL*, the *Traffic Mode Type* continues to describe three general distribution algorithms: Override, Load-share and Broadcast. The change in the behavior of the SGP when selecting an ASP for traffic distribution introduced by *LOADSEL* is that the SGP also takes into account the concept of a *Load Selector*. The *LOADSEL* procedures are summarized as follows:

Override:– When distributing messages to an Override Application Server, the SGP first determines the *Load Selector* associated with the message. The SGP then distributes the message to the ASP that is active for the traffic flow indicated by the *Load Selector* within the AS. In an Override Application Server, at most one ASP can be active for the AS within a *Load Selector* at any given point in time. The active ASP associated with the *Load Selector* for the AS is used.

Load-share:– When distributing messages to a Load-share Application Server, the SGP first determines the *Load Selector* associated with the message. The SGP then selects one of the active ASPs associated with the *Load Selector* within the AS using an implementation dependent load-sharing algorithm based on some unspecified aspect of the traffic or static configuration data.

Broadcast:– When distributing messages to a Broadcast Application Server, the SGP first determines the *Load Selector* associated with the message. The SGP then selects all of the active ASPs associated with the *Load Selector* within the AS and sends a copy of the message to each ASP. (The ASPs themselves decide which ASP(s) will process the message.)

The result of this extension is that Application Server Processes have control over their placement within an Application Server and can control the traffic that they receive by registering and activating for specific *Load Selectors* within the AS.

2. Conventions

The keywords **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT, SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **NOT RECOMMENDED**, **MAY**, and **OPTIONAL**, when they appear in this document, are to be interpreted as described in [RFC 2119].

3. Protocol Elements

The following subsections describe the parameters which are added by this extension, their format and the messages in which they are used.

3.1. Parameters

LOADSEL adds one new parameter: the *Load Selector* parameter.

3.1.1. Load Selector

The *Load Selector* parameter is used in the *REG RSP*, *ASPAC*, *ASPAC ACK*, *ASPIA*, *ASPIA ACK*, and *NTFY* messages. It is used to identify the placement of the ASP within an Application Server. It identifies the Load Selection (partition of traffic flow) for which the ASP is registering, activating or deactivating, or when the SG is indicating or notifying of an ASP state change.

The *Load Selector* parameter is formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag = 0x0018         |             Length            |
+-------------------------------+-------------------------------+
\                                                               \
/                       Load Selector  1                        /
\                                                               \
+---------------------------------------------------------------+
\                                                               \
/                             ...                               /
\                                                               \
+---------------------------------+-----------------------------+
\                                                               \
/                       Load Selector  n                        /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0018`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Selector* parameter contains the following fields:

Load Selector field: n x 32-bits (unsigned integer)

The *Load Selector* forms and extension to the *Routing (Link) Key* and an attribute of the Application Server as indicated by the *Routing Context* (*Interface Identifier(s)*).

The *Load Selector* field is an identifier, that **MUST** be unique within an Application Server and **MAY** be unique within an administrative domain, that identifies the placement or Load Selection of an ASP within an AS. This identifier, in conjunction with any Application Server identifier (i.e, *Routing Context* or *Interface Identifier(s)*), identifies the traffic flow for which an ASP is registering, activating or deactivating, or for which an SG is providing notification of an ASP or AS state change.

When the *Load Selector* parameter is included in the *ASPAC*, *ASPAC ACK*, *ASPIA*, *ASPIA ACK* or *NTFY* message, one *Routing Context* (*Interface Identifier(s)*) representing a single Application Server **SHOULD** be associated (specified or implied) with the message.

3.1.2. Load Selection

The *Load Selection* parameter is used in the *REG REQ* message. It is used to register the placement of an ASP within an Application Server.

The *Load Selection* parameter is formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag = 0x0019         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                     Load Key Parameter(s)                     /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0019`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Key Parameter(s)* can contain the following parameters:

UA	**Parameters**		
M2UA	SLS Range	Mandatory	*1
M3UA	Service Indicators	Conditional	*2
	CIC Range	Conditional	
SUA	Address Range	Conditional	*3
	DLR Label	Conditional	
	TID Label	Conditional	

ISUA	CIC Range	Mandatory	
TUA	Address Range	Condtional	*3
	Transaction Id Range	Conditional	*4

Note 1: The *SLS Range* parameter has not yet been defined for M2UA [M2UA].

Note 2: When the *Service Indicators* parameter is used in the *Load Selection*, it **SHOULD** be more restricted than (a subset of) any *Service Indicators* parameter present in the *Routing Key* parameter for routing purposes.

Note 3: When the *Address Range* parameter is used in the *Load Selection*, it **SHOULD** be more restrictive than (a subset of) any *Address Range* parameter present in the *Routing Key* parameter for routing purposes.

Note 4: When the *Transaction Id Range* parameter is used in the *Load Selection*, it **SHOULD** be more restrictive than (a subset of) any *Transaction Id Range* parameter present in the *Routing Key* parameter for routing purposes.

The *Load Selection* parameter **SHOULD** contain at least one Load Selection Parameter.

3.2. Messages

LOADSEL adds the *Load Selector* parameter as an **OPTIONAL** parameter to be used in conjunction with the common *Traffic Mode Type* in the following messages: [2]

REG REQ	Registration Request message
REG RSP	Registration Response message
ASPAC	ASP Active message
ASPAC ACK	ASP Active Ack message
ASPIA	ASP Inactive message
ASPIA ACK	ASP Inactive Ack message
NTFY	Notify message

3.2.1. Registration Request (REG REQ)

LOADSEL supplements the *Registration Request (REG REQ)* message by permitting the following optional parameters to be included in the *Routing Key* [M3UA...TUA] or *Link Key* [M2UA] parameter within the message:

Extension Parameters	
Load Selection	Optional

The *Load Selector* parameter is used in the *Routing Key* (*Link Key*) to further refine the traffic load selection to be received by the registering ASP.

The format of the resulting *Routing Key* or *Link Key* parameter is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag =                |           Length = 8          |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                   Local-RK(LK)-Identifier                     |
+-------------------------------+-------------------------------+
|          Tag = 0x0000         |           Length = 8          |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                      Traffic Mode Type                        |
+-------------------------------+-------------------------------+
|          Tag = 0x0019         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                       Load Selection                          /
\                                                               \
+---------------------------------------------------------------+
\                                                               \
/                       Key Parameter(s)                        /
\                                                               \
/                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0019`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

When an ASP wishes to register within a Load Selection associated with an Application Server, it includes the *Load Selection* parameter in the *Routing Key* (*Link Key*) for that Application Server in the *REG REQ* message.

The *Load Selection* parameter indicates the traffic load selection to be used within the *Application Server* as identifier by the *Routing (Link) Key* parameter.

When the ASP includes the *Load Selection* parameter in the *Routing (Link) Key* it expects the SGP to respond with a *Load Selector* parameter in addition to the *Routing Context* to indicate the identifier for the *Load Selection* specified. In this way, the *Load Selection* parameter is analogous to the *Routing Key* and the *Load Selector* parameter is analogous to the *Routing Context*.

No other changes to the *REG REQ* message, *Routing Key* or *Link Key* parameters formats are provided by this extension.

3.2.2. Registration Response (REG RSP)

LOADSEL supplements the *Registration Response (REG RSP)* message by permitting the following optional parameters to be include in the *Registration Result* parameter within the message:

Extension Parameters	
Load Selector	Optional

The *Load Selector* parameter is used in the *Regsitration Result* to provide an identifier for the registered *Load Selection*.

The format of the resulting *Registration Response* parameter is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag =                |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                     Local-RK(LK)-Identifier                   |
+---------------------------------------------------------------+
|                       Regsitration Status                     |
+---------------------------------------------------------------+
|              Routing Context (Interface Identifier)           |
+---------------------------------------------------------------+
|                           Load Selector                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

When the SGP replies to a *REG REQ* that contained a *Load Selection* parameter, and the registration is successful, it returns the *Load Selector* in the *REG RSP* associated with the *Load Selection* in the *REG REQ*.

When the registration is unsuccessful (i.e. the *Registration Status* indicates an error), the *Load Selector* **SHOULD NOT** be included in the *REG RSP* message.

LOADSEL extends the common *Registration Status* parameter in the *REG RSP* message by adding the following values to the *Registration Status*:

Value	**Description**
12	Error – Unsupported Load Selection
13	Error – Invalid Load Selection
14	Error – Cannot Support Unique Loadsharing
15	Error – Unsupported LS Parameter Field

EDITOR'S NOTE:- The *Registration Status* values shown as `12`, `13`, `14` and `15` above will be assigned by IANA as a value of the UA-specific *Registration Status* parameter for each SIGTRAN UA and may change its value in further versions of this document.

No other changes to the *REG RSP* message or *Regsitration Response* parameter formats are provided by this extension.

3.2.3. ASP Active (ASPAC)

LOADSEL supplements the *ASP Active (ASPAC)* message by permitting the following optional parameters to be included in the *ASPAC* message:

Extension Parameters	
Load Selector	Optional

The format of the resulting *ASPAC* message is as follows: [2]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag = 0x000b         |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                       Traffic Mode Type                       |
+---------------------------------------------------------------+
\                                                               \
/                        Routing Context                        /
\                              or                               \
/                    Interface Identifier(s)                    /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0018         |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                         Load Selector                         /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0004         |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0018`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Selector* parameter is used by the ASP in the *ASPAC* message to indicate the range of traffic for which the ASP is activating. The Application Servers for which the *Load Selectors* apply is either indicated in the *ASPAC* message by providing the associated *Routing Context* (*Interface Identifier(s)*) or, if there is no *Routing Context* (*Interface Identifier*) parameter in the *ASPAC* message, the associated Application Servers are implied by SGP and ASP configuration data. (See Section 4.1.5 – "ASP Active Procedures".)

No other changes to the *ASPAC* message format are provided by this extension.

3.2.4. ASP Active Ack (ASPAC ACK)

LOADSEL supplements the *ASP Active Ack (ASPAC ACK)* message by permitting the following optional parameters to be included in the *ASPAC ACK* message:

Extension Parameters	
Load Selector	Optional

The format of the resulting *ASPAC ACK* message is as follows: [2]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag = 0x000b         |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                       Traffic Mode Type                       |
+---------------------------------------------------------------+
\                                                               \
/                        Routing Context                        /
\                              or                               \
/                    Interface Identifier(s)                    /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0018         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Load Selector                        /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0004         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0018`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Selector* parameter is used by the SGP in the *ASPAC ACK* message to indicate the range of traffic for which the SGP has activated the ASP. The Application Servers for which the *Load Selectors* apply is either indicated in the *ASPAC ACK* message by providing the associated *Routing Context* (*Interface Identifier(s)*) or, if there is no *Routing Context* (*Interface Identifier*) parameter in the *ASPAC ACK* message, the associated Application Servers are implied by SGP and ASP configuration data. (See Section 4.1.5 – "ASP Active Procedures".)

No other changes to the *ASPAC ACK* message format are provided by this extension.

3.2.5. ASP Inactive (ASPIA)

LOADSEL supplements the *ASP Inactive (ASPIA)* message by permitting the following parameters to be included in the *ASPIA* message:

Extension Parameters	
Load Selector	Optional

The format of the resulting *ASPIA* message is as follows: [2]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                        Routing Context                        /
\                              or                               \
/                    Interface Identifier(s)                    /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0018         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                         Load Selector                         /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0004         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0018`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Selector* parameter is used by the ASP in the *ASPIA* message to indicate the range of traffic for which the ASP is deactivating. The Application Servers for which the *Load Selectors* apply is either indicated in the *ASPIA* message by providing the associated *Routing Context* (*Interface Identifiers*) or, if there is no *Routing Context* (*Interface Identifier*) parameter in the *ASPIA* message, the associated Application Servers are implied by SGP and ASP configuration data. (See Section 4.1.6 – "ASP Inactive Procedures".)

No other changes to the *ASPIA* message format are provided by this extension.

3.2.6. ASP Inactive Ack (ASPIA ACK)

LOADSEL supplements the *ASP Inactive Ack (ASPIA ACK)* message by permitting the following parameters to be included in the *ASPIA ACK* message:

Extension Parameters	
Load Selector	Optional

The format of the resulting *ASPIA ACK* message is as follows: [2]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                        Routing Context                        /
\                              or                               \
/                    Interface Identifier(s)                    /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0018         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                         Load Selector                         /
\                                                               \
+-------------------------------+-------------------------------+
|          Tag = 0x0004         |             Length            |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0018`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Selector* parameter is used by the SGP in the *ASPIA ACK* message to indicate the range of traffic for which the SGP has deactivated the ASP. The Application Servers for which the *Load Selectors* apply is either indicated in the *ASPIA ACK* message by providing the associated *Routing Context* (*Interface Identifiers*) or, if there is no *Routing Context* (*Interface Identifier*) parameter in the *ASPIA ACK* message, the associated Application Servers are implied by SGP and ASP configuration data. (See Section 4.1.6 – "ASP Inactive Procedures".)

No other changes to the *ASPIA ACK* message format are provided by this extension.

3.2.7. Error (ERR)

LOADSEL supplements the *Error (ERR)* message by adding the following values to the common mandatory *Error Code* parameter in the *ERR* message:

29 Invalid Load Selector

EDITOR'S NOTE:- The *Error Code* value shown as `29`) above will be assigned by IANA as a value of the common *Error Code* parameter for SIGTRAN UAs and may change its value in further versions of this document.

These new error codes are interpreted as follows: [2]

The *"Invalid Load Selector"* error would be sent in an *ERR* message if the SG determines that one or more *Load Selectors* in the *Load Selector* parameter provided by the ASP is invalid, is not configured, or cannot be supported.

No other changes to the *ERR* message or *Error Code* parameter format are provided by this extension. See Section 4 for extension procedures associated with the *ERR* message.

3.2.8. Notify (NTFY)

LOADSEL supplements the *Notify (NTFY)* message by permitting the following parameters to be included in the *NTFY* message:

Extension Parameters	
Load Selector	Optional

The format of the resulting *NTFY* message is as follows: [2]

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x000d          |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|          Status Type          |      Status Information       |
+-------------------------------+-------------------------------+
|         Tag = 0x0012          |          Length = 8           |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
|                        ASP Identifier                         |
+---------------------------------------------------------------+
\                                                               \
/                        Routing Context                        /
\                              or                               \
/                    Interface Identifier(s)                    /
\                                                               \
+-------------------------------+-------------------------------+
|         Tag = 0x0018          |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                         Load Selector                         /
\                                                               \
+-------------------------------+-------------------------------+
|         Tag = 0x0004          |            Length             |
+- - - - - - - - - - - - - - - -+- - - - - - - - - - - - - - - -+
\                                                               \
/                          Info String                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EDITOR'S NOTE:- The parameter tag values shown as `0x0018`) above will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

The *Load Selector* parameter is used by the SGP in the *NTFY* message to indicate the traffic load positions which led to or have contributed to the change in state of an associated Application Server when sending a *NTFY* message that indicates and Application Server state change. The Application Servers for which the *Load Selectors* apply is either indicated in the *NTFY* message by providing the associated *Routing Context* (*Interface Identifiers*) or, if there is no *Routing Context* (*Interface Identifier*) parameter in the *NTFY* message, the associated Application Servers are implied by SGP and ASP configuration data. (See Section 4.1.7 – "Notification Procedures".)

No other changes to the *NTFY* message format are provided by this extension.

4. Procedures

LOADSEL provides for an additional level of control over the traffic distribution patterns within an Application Server. *LOADSEL* provides the *Load Selector* parameter which may be optionally included in the *ASPAC*, *ASPAC ACK*, *ASPIA*, *ASPIA ACK* and *NTFY* messages. In addition, it provides procedures for use of the *Load Selector* parameter in association with these messages.

4.1. AS and ASP State Maintenance

In addition to the SGP maintaining the state of each remote ASP in each Application Server that the ASP is configured to receive traffic, under *LOADSEL*, SGP **MAY** also maintain the state of each remote ASP in each Load Selection within an Application server that the ASP is configured to receive traffic. Aside from the procedures described in Section 4.1.7 "Notification Procedures", no management procedures are provided by *LOADSEL* for maintaining the state of a Load Selection within an Application Server. The Load Selection state is maintained separate from the ASP and AS states.

4.1.1. ASP State

LOADSEL uses the existing UA [IUA...TUA] definitions and procedures with regard to ASP State.

4.1.2. AS State

The state of the Application Server is maintained in the Signalling User Adaptation Layer on the SGPs. The state of the Application Server changes due to ASP state transitions. The possible states of an Application Server using *LOADSEL* are:

AS-DOWN: The Application Server is unavailable. This state implies that all related ASPs are in the **ASP-DOWN** state for this AS. Initially, the AS will be in this state. An Application Server is in the **AS-DOWN** state when it is removed from a configuration.

AS-INACTIVE: The Application Server is available but no application traffic is active (i.e, one or more related ASPs are in the **ASP-INACTIVE** state, but there is no ASP in the **ASP-ACTIVE** state). The recovery timer T(r) is not running or has expired.

AS-ACTIVE: The Application Server is available and application traffic is active. This state implies that there is at least one ASP in the **ASP-ACTIVE** state within the AS.

AS-PENDING: An active ASP has transition to **ASP-INACTIVE** or **ASP-DOWN** and it was the last remaining active ASP for a Load Selection in the AS. A recovery timer T(r) **SHOULD** be started and all incoming signalling messages **SHOULD** be queue by the SGP for each of the Load Selections for which there is no ASP in the **ASP-ACTIVE** state. If an ASP becomes **ASP-ACTIVE** for each Load Selection in the AS before T(r) expires, the AS is moved to the **AS-ACTIVE** state and all the queued messages for the Load Selection will be sent to the newly active ASPs on each Load Selection.

4.1.3. ASP Up Procedures

When an SGP receives an ASP Up messages from an ASP and the ASP is configured at the SGP in one or more Load Selections associated with one or more Application Servers, if the ASP is not already in the **ASP-INACTIVE** state for the associated Application Servers, the SGP moves the ASP into the **ASP-INACTIVE** state for the configured Load Selections within the associated Application Servers. In any event, the SGP responds with an *ASPUP ACK* message.

If the ASP transition to the **ASP-INACTIVE** state within these Load Selections results in a state change in the associated Application Servers, the SGP moves the Application Server to the **AS-INACTIVE** state. If the ASP transition to the **ASP-INACTIVE** state for the given Load Selection within an associated Application Server results in a change in state of the AS (or the Load Selections associated with the AS), the SGP also follows the "Notification Procedures" (described in Section 4.1.7) for the Application Server.

Otherwise, the "ASP Up Procedures" described by the UAs [3] apply also to *LOADSEL*.

4.1.4. ASP Down Procedures

When an SGP receives an ASP Down messages from an ASP and the ASP is configured at the SGP in one or more Load Selections associated with one or more Application Servers, if the ASP is not already in the **ASP-DOWN** state for the associated Application Servers, the SGP moves the ASP to the **ASP-DOWN** state for the configured Load Selections within the associated Application Servers. In any event, the SGP responds with an ASP Down Ack.

If the ASP transition to the **ASP-DOWN** state within these Load Selections results in a state change in the associated Application Servers, the SGP moves the Application Server to the **AS-DOWN** state.

Otherwise, the "ASP Down Procedures" described by the UAs [3] apply also to *LOADSEL*.

4.1.5. ASP Active Procedures

When an ASP wishes to activate for a set of Load Selections associated with an Application Server, it includes the *Load Selector* parameter in the *ASPAC* message.

When an SGP receives and *ASPAC* message requesting activation for a set of Load Selections within an AS or the SG otherwise activates the ASP for a set of Load Selections within an AS, the SG sends an *ASPAC ACK* message including the *Load Selector* parameter indicating the Load Selections for which the ASP has been activated. If the SGP is responding to an ASP that included the *Load Selector* parameter in the *ASPAC* message, the SG **MUST** include the *Load Selector* parameter in the response *ASPAC ACK* message.

If the *Load Selector* included in an *ASPAC* message contains invalid information or indicate an unsupported Load Selection, or a *Load Selector* parameter required by the SGP is missing, the SGP replies to the *ASPAC* message with an *ERR*("Invalid Load Selector") message and takes no further action with regard to AS or ASP state.

The Application Servers associated with the *Load Selector* is indicated in the *ASPAC (ACK)* message by *Routing Context* (*Interface Identifiers*) or, when the *Routing Context* (*Interface Identifier*) parameter is missing from the *ASPAC (ACK)* message, is implied by ASP and SGP configuration data. When the *Load Selector* parameter is included in the *ASPAC (ACK)* message, the *Routing Context* (*Interface Identifier*) parameter or implicit configuration data **SHOULD** be associated with a single Application Server.

If the ASP transition to the **ASP-ACTIVE** state for the given Load Selections within an associated Application Server results in a change in state of the AS (or the Load Selections associated with the AS), the SGP follows the "Notification Procedures" (described in Section 4.1.7) for the Application Server.

Otherwise, the "ASP Active Procedures" described by the UAs [3] apply also to *LOADSEL*.

4.1.6. ASP Inactive Procedures

When an ASP wishes to deactivate for a set of Load Selections associated with an Application Server, it includes the *Load Selector* parameter in the *ASPIA* message.

When an SGP receives an *ASPIA* message requesting deactivation for a set of Load Selections within an AS or the SG otherwise deactivates the ASP for a set of Load Selections within an AS, the SG sends an *ASPIA ACK* message including the *Load Selector* parameter for which the ASP has been deactivated. If the SGP is responding to an ASP that included the *Load Selector* parameter in the *ASPIA* message, the SG **MUST** include the *Load Selector* parameter in the response *ASPIA ACK* message.

If the SGP receives an *ASPIA* message from an ASP that is active for a set of Load Selections associated with the Application Server for which the ASP is requesting deactivation, and the *Load Selector* parameter is not present in the *ASPIA* message, the SGP will interpret this as a request to deactivate the ASP for all the Load Selections associated with the Application Server for which the ASP is active.

If the *Load Selector* included in an *ASPAC* message contains invalid information or indicates an unsupported *Load Selector*, the SGP replies to the *ASPAC* message with an *ERR*("Invalid Load Selector") message and takes no further action with regard to AS or ASP state.

The Application Servers associated with the *Load Selector* is indicated in the *ASPIA (ACK)* message by *Routing Context* (*Interface Identifiers*) or, when the *Routing Context* (*Interface Identifier*) parameter is missing from the *ASPIA (ACK)* message, is implied by ASP and SGP configuration data. When the *Load Selector* parameter is included in the *ASPIA (ACK)* message, the *Routing Context* (*Interface Identifier*) parameter or implicit configuration data **SHOULD** be associated with a single Application Server.

If the ASP transition to the **ASP-INACTIVE** state for the given Load Selections within an associated Application Server results in a change in state of the AS (or the Load Selections associated with the AS), the SGP follows the "Notification Procedures" (described in Section 4.1.7) for the Application Server.

Otherwise, the "ASP Inactive Procedures" described by the UAs [3] apply also to *LOADSEL*.

4.1.7. Notify Procedures

4.1.7.1. AS State Change

When an ASP makes a state transition and is configured for a set of Load Selections in one or more Application Servers, the ASP state transition may result in a change to the state of the associated Application Servers, or Load Selections within those AS, at the SGP. When a recovery timer T(r) expires, the associated Application Server will transition from the **AS-PENDING** state to the **AS-INACTIVE** state.

Whenever an AS changes state, or the condition of the AS Load Selections perpetuating the current AS state changes, the SGP **MUST** notify all ASPs not in the **ASP-DOWN** state configured for the Application Server by sending a *Notify (NTFY)* message to each indicating the state of the Application Server in the *Status Information* field of the *Status* parameter in the *NTFY* message.

When an Application Server is configured for *LOADSEL*, the SGP **MUST** also include a list of Load Selections in the *Load Selector* parameter for which the Application Server is configured that caused the AS state change or is perpetuating the AS state.

The SGP includes the *Routing Context* (*Interface Identifiers*) that identifies the Application Server for which the *NTFY* message is being sent. If, however, the ASP is not configured for more than one Application Server, the *Routing Context* (*Interface Identifier*) **MAY** be excluded from the *NTFY* message as it is implied by configuration data.

Otherwise, the "Notification Procedures" described by the UAs [3] apply also to *LOADSEL*.

Examples are given in Section 5.

4.1.7.2. ASP Override

Whenever an ASP becomes active for a Load Selection in an Override Application Server, the *Load Selector* **MUST** be placed in a *NTFY* message with a *Status Information* indicating "Alternate ASP active in AS", along with the *Routing Context* (*Interface Identifiers*) for the associated AS and any *ASP Identifier* associated with the ASP for which the notification is being given.

Otherwise, the "Notification Procedures" described by the UAs [3] apply also to *LOADSEL*.

4.1.8. Registration Procedures

LOADSEL provides the ability for the ASP to obtain a *Load Selector* value for a given range of traffic (*Load Selection*) using dynamic registration.

When the ASP wishes to register a *Load Selector* as part of the normal registration procedure, it includes the *Load Selection* parameter in the *REG REQ* message to the SGP. The *Load Selection* parameter describes the range of traffic load for which the ASP wishes to obtain a *Load Selector* value.

When an SGP receives a *REG REQ* message with a *Load Selection* parameter, it will determine whether the *Load Selection* parameter describes a non-overlapping load selection and then, if the registration of the associated *Routing Key* is successful, it assigns a *Load Selector* to the traffic range and returns the value in the *Load Selector* parameter in the *REG RSP* message.

In addition to the normal registration procedures of the UAs, the following additional error conditions can occur:

"Error − Unsupported Load Selection"

> This error **MUST** be sent in an **Registration Response (REG RSP)** message whenever the SG determines that the *Load Selection* provided in a **REG REQ** message has not been configured and the SG does not support dynamic allocation of *Load Selectors* for the specified Key, and **MUST** be sent in an **Registration Response (REG RSP)** message whenever the SG determines that the *Load Selection* provided in a **REG REQ** message is not supported by the SG.

"Error − Invalid Load Selection."

> If the SGP determines that the received *Load Selection* data is invalid, or contains invalid parameter values, the SGP returns a *Registration Response (REG RSP)* message to the ASP containing a *Registration Result* "Error − Invalid Load Selection".

"Error − Cannot Support Unique Loadsharing."

> If the SGP determines that a unique *Load Selection* cannot be created, the SGP returns a *Registration Response (REG RSP)* message to the ASP, with a *Registration Status* of "Error − Cannot Support Unique Loadsharing." An incoming signalling message received at an SGP should not match against more than one *Load Selection*.

"Error − Unsupported LS Parameter Field."

> If the SGP determines that one or more of the *Load Selection* parameters are not supported for the purpose of creating new Load Selection entries, the SGP returns a *Registration Response (REG RSP)* message to the ASP, containing a *Registration Result* "Error − Unsupported LS Parameter Field". This result **MAY** be used if, for example, the SGP does not support the *CIC Range* parameter.

Otherwise, the "Registration Procedures" described by the UAs [3] apply also to *LOADSEL*.

4.2. Interworking Procedures

LOADSEL supports the ASP Extension procedures described in ASPEXT [ASPEXT] and defines the following *ASP Extension* value:

1 LOADSEL Extension [LOADSEL]

The following procedures may be used where the *ASPEXT* procedures are not used:

Whenever an ASP receives an *ASPIA ACK* not containing a *Load Selector* parameter in response to an *ASPAC* containing the parameter, the ASP will assume that the the SGP or IPSP does not support *LOADSEL* and **MUST** fall back to the non-Load Selection UA procedures.

Whenever an ASP receives an *ERR* message or an unsuccessful *REG RSP* message indicating a problem with the *Load Selection* parameter in the *REG REQ* message, the ASP will assume that the SGP or IPSP does not support *LOADSEL* and **MUST** fall back to the non-Load Selection UA procedures.

5. Examples

Figure 3 illustrates the example configuration that is used for all the examples in this section. The example configuration consist of:

- Two SGs (SG1 and SG2) acting as STPs in the SS7 network and consisting (for example) of a single SGP. Each SG is connected to each of the ASPs in the example configuration.
- Four ASPs (ASP1, ASP2, ASP3 and ASP4). Each ASP is connected to both of the SGs in the example configuration.
- Two Load Selections (LS1 and LS2) are associated with the Application Server. The traffic that corresponds to each Load Selections is different in each example.

5.1.1. Initialization

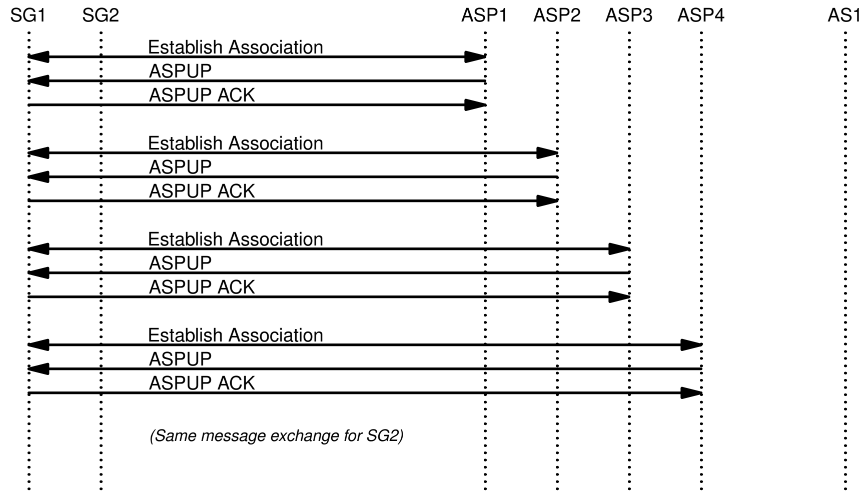

Figure 4. Example – Initialization

Figure 4 illustrates the common initialization procedure use for all of the examples. Each ASP establishes an SCTP Association with SG1 and sends an *ASPUP* message to which it receives and ASP Up response. The ASPs are not statically configured to serve specific AS or LS within the AS, so no Notify messages are received. The same sequence of messages are also exchanged with SG2.

5.2. M3UA with Override AS and Load Selection based on CIC

This example is for an M3UA [M3UA] configuration with the Application Server (AS1) configured with a Traffic Mode Type of Override. The Application Server (AS1) has associated with it a Routing Key (RK1) that consists of a Destination

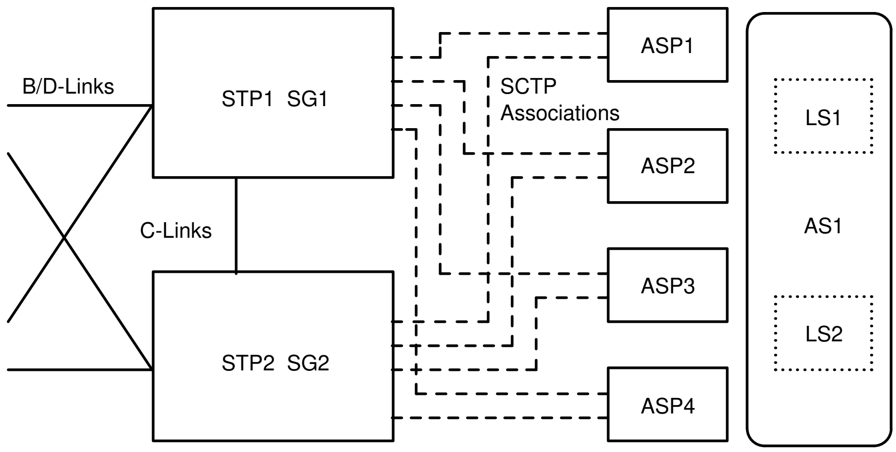

Figure 3. Example Configuration

Point Code that corresponds to the AS1 (MGC1) point code (PC1), an Originating Point Code that corresponds to a remote MGC2 point code (PC2), and the SI value for ISUP (SI=5). The Load Selections (LS1 and LS2) correspond to two sets of CIC values which correspond to two different trunk groups between MGC1 and MGC2 (TG1 and TG2).

5.2.1. Activation

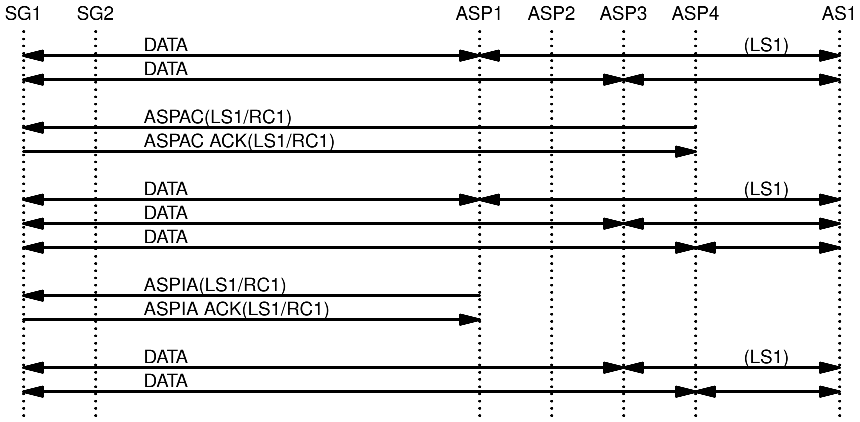

Figure 5. M3UA Example – Activation

Figure 5 illustrates activation of the ASPs for Load Selections within the Override Application Server. The sequence of events is as follows:

(1) ASP1 sends an *ASPAC* message to SG1 identifying Load Selection LS1 within Application Server AS1/RC1 and receives an acknowledgment and a notification. Data is transferred between the SG and ASP1 for Load Selection LS1 within AS1.

(2) ASP2 sends an *ASPAC* message to SG1 identifying Load Selection LS2 within Application Server AS1/RC1 and receives an acknowledgment and a notification. ASP1 also receives notification that AS1/RC1 is active for LS2. Data is transferred between the SG and ASP2 for Load Selection LS2 within AS1.

(3) ASP3 sends an *ASPIA* message to SG1 identifying Load Selection LS1 within Application Server AS1/RC1 and receives an acknowledgment and a notification.

(4) ASP4 sends an *ASPIA* message to SG1 identifying Load Selection LS1 and LS2 within Application Server AS1/RC1 and receives an acknowledgment and a notification.

(5) The same exchange is repeated for SG2.

5.2.2. Failure of ASP1

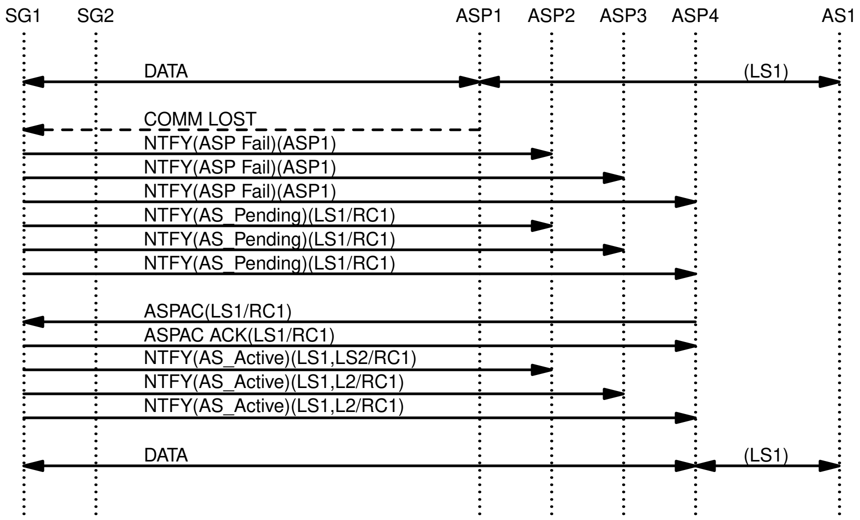

Figure 6. M3UA Example – Failure of ASP1

Figure 6 illustrates the failure of ASP1. The sequence of events is as follows:

(1) Data for LS1 within AS1 is exchanged between SG1 and ASP1. Data for LS2 within AS1 is exchanged between SG1 and ASP2.

(2) Communication is lost between SG1 and ASP1. ASP2, ASP3, and ASP4 are notified of the failure of ASP1 and the change of state of AS1 to **AS-PENDING** for LS1. Data for LS2 in AS1 is unaffected.

(3) ASP4 (spare) responds to the **AS-PENDING** notification and activates for LS1 in AS1/RC1. ASP2, ASP3 and ASP4 receive an **AS-ACTIVE** notification. Data for LS1 in AS1 is now exchanged with ASP4.

5.2.3. Sparing

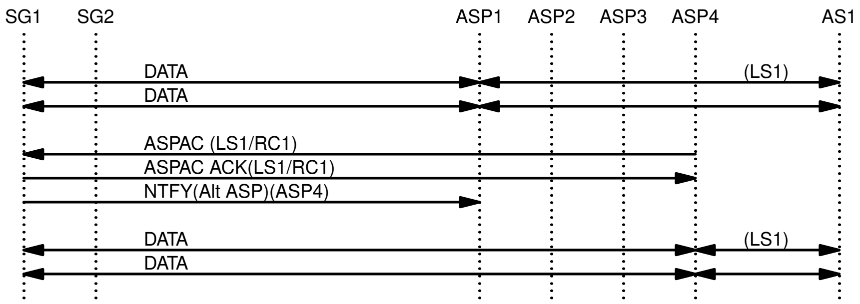

Figure 7. M3UA Example – Sparing

Figure 7 illustrates a sparing situation where one ASP takes over traffic from another so that the original ASP can be taken out of service. The sequence of events is as follows:

(1) Data for LS1 in AS1 is exchange between SG1 and ASP1.

(2) ASP4 (spare) activates for LS1 in AS1 and receives and acknowledgment. ASP4 has overridden ASP1 and a notification is sent to ASP1 that indicates that ASP4 in now the "Alternate ASP Active for AS".

(3) Data for LS1 in AS1 is now being exchanged between SG1 and ASP4.

(4) ASP1 can now be taken down and out of service.

5.3. SUA with Load-share AS and Load Selection based on GT

This example is for an SUA [SUA] configuration with the Application Server (AS1) configured with a Traffic Mode Type of Load-share. The Application Server (AS1) has associated with it (RK1) that consists of Destination Point Code and Subsystem Number that corresponds to the AS1 (HLR1) point code (PC1). The Load Selections (LS1 and LS2) correspond to two sets of Global Titles which correspond to Mobile and GSTN numbering.

5.3.1. Activation

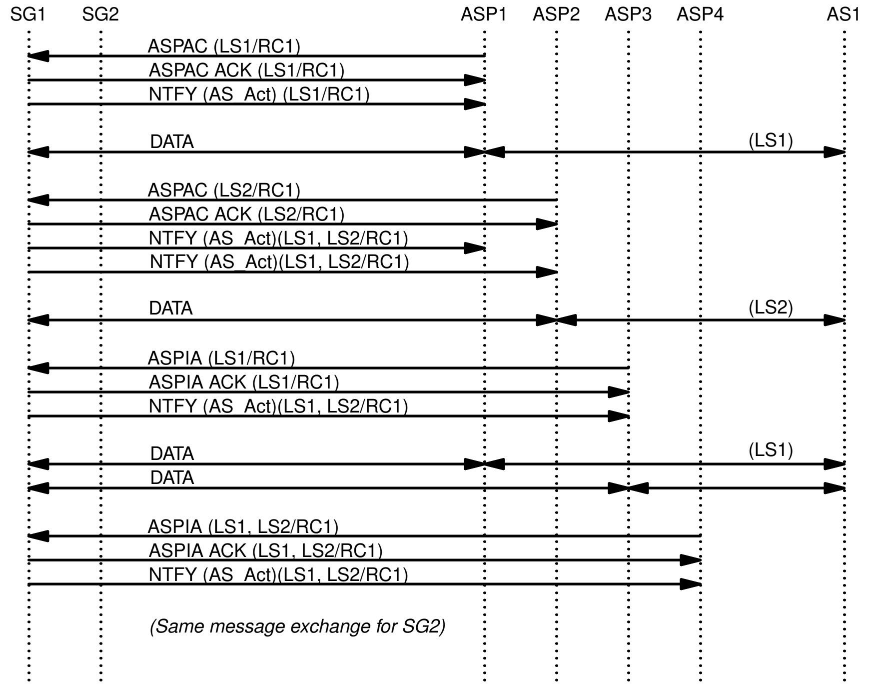

Figure 8. SUA Example – Activation

Figure 8 illustrates activation of the ASPs for Load Selections within the Load-share Application Server. The sequence of events is as follows:

(1) ASP1 sends an *ASPAC* message to SG1 identifying Load Selection LS1 within Application Server AS1/RC1 and receives an acknowledgment and a notification. Data is transferred between the SG and ASP1 for Load Selection LS1 within AS1.

(2) ASP2 sends an *ASPAC* message to SG1 identifying Load Selection LS2 within Application Server AS1/RC1 and receives an acknowledgment and a notification. ASP1 also receives notification that AS1/RC1 is active for LS2. Data is transferred between the SG and ASP2 for Load Selection LS2 within AS1.

(3) ASP3 sends an *ASPAC* message to SG1 identifying Load Selection LS1 within Application Server AS1/RC1 and receives an acknowledgment and a notification. Data is load-shared between the SG and ASP1 and ASP3 for Load Selection LS1 within AS1.

(4) ASP4 sends an *ASPAC* message to SG1 identifying Load Selection LS1 and LS2 within Application Server AS1/RC1 and receives an acknowledgment and a notification.

(5) The same exchange is repeated for SG2.

5.3.2. Failure of ASP1 and ASP2

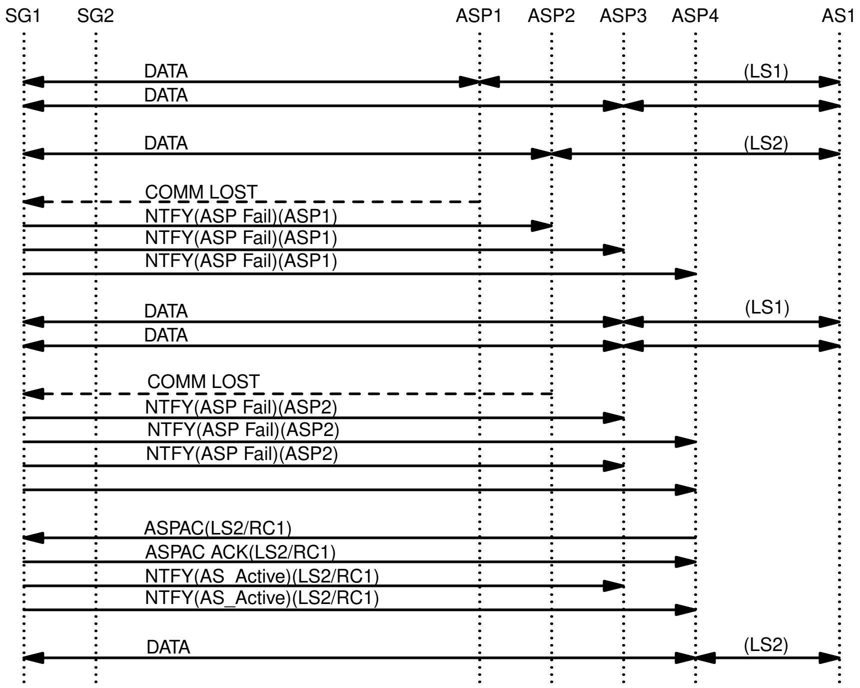

Figure 9. SUA Example – Failure of ASP1 and ASP2

Figure 9 illustrates the failure of ASP1 followed by the failure of ASP2. The sequence of events is as follows:

(1) Data for LS1 within AS1 is load-shared between ASP1 and ASP3. Data for LS2 within AS1 is exchanged with ASP2.

(2) Communication is lost between SG1 and ASP1. ASP2, ASP3, and ASP4 are notified of the failure of ASP1. Data for LS1 in AS1 is directed toward ASP3 only. Data for LS2 in AS1 is unaffected.

(3) Communication is lost between SG1 and ASP2. ASP3 and ASP4 are notified of the failure of ASP1 as well of the AS1 state change to **AS-PENDING**. Data for LS2 is queued at the SG.

(4) ASP4 (spare) responds to the **AS-PENDING** notification and activates for LS2 in AS1/RC1. ASP3 and ASP4 receive an **AS-ACTIVE** notification. Data for LS2 in AS1 is now exchanged with ASP4.

5.3.3. Sparing

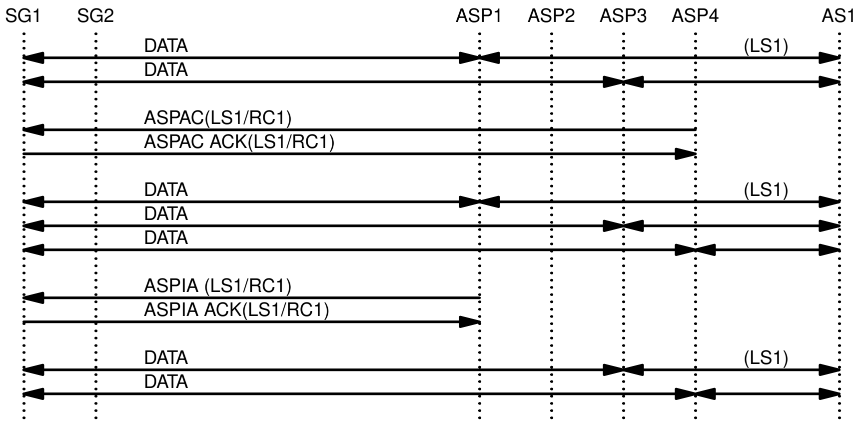

Figure 10. SUA Example – Sparing

Figure 10 illustrates a sparing situation where one ASP takes over traffic from another so that the original ASP can be taken out of service. The sequence of events is as follows:

(1) Data for LS1 in AS1 is load-shared between SG1 and ASP1 and ASP3.

(2) ASP4 (spare) activates for LS1 in AS1 and receives and acknowledgment. Data for LS1 in AS1 is now being load-shared between SG1 and ASP1, ASP3 and ASP4.

(3) ASP1 deactivates for LS1 in AS1 and receives and acknowledgment but no notification.

(4) Data or LS1 in AS1 is now load-shared between SG1 and ASP3 and ASP4.

(5) ASP1 can now be taken down and out of service.

5.4. TUA with Broadcast AS and Load Selection based on DID

This example is for an TUA [TUA] configuration with the Application Server (AS1) configured with a Traffic Mode Type of Broadcast. The Application Server (AS1) has associated with it (RK1) that consists of Destination Point Code and Subsystem Number that corresponds to the AS1 (HLR1) point code (PC1). The Load Selections (LS1 and LS2) correspond to two sets of Dialog Ids which correspond to even and odd Dialog Ids.

5.4.1. Activation

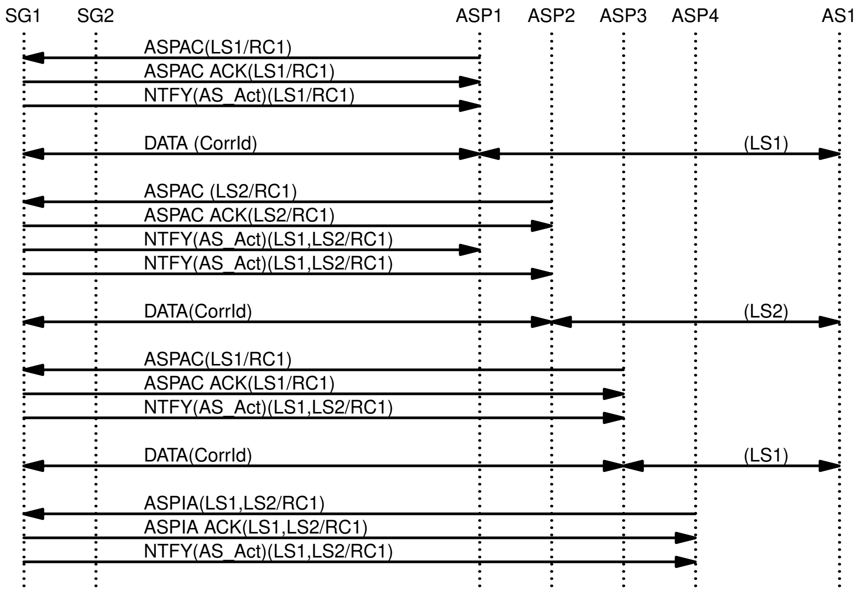

Figure 11. TUA Example − Activation

Figure 11 illustrates activation of the ASPs for Load Selections within the Broadcast Application Server. The sequence of events is as follows:

(1) ASP1 sends an *ASPAC* message to SG1 identifying Load Selection LS1 within Application Server AS1/RC1 and receives an acknowledgment and a notification. Data is transferred between the SG and ASP1 for Load Selection LS1 within AS1. The initial Data Messages for LS1 within AS1 are tagged with a *Correlation Id*.

(2) ASP2 sends an *ASPAC* message to SG1 identifying Load Selection LS2 within Application Server AS1/RC1 and receives an acknowledgment and a notification. ASP1 also receives notification that AS1/RC1 is active for LS2. Data is transferred between the SG and ASP2 for Load Selection LS2 within AS1. The initial Data Messages for LS2 within AS1 are tagged with a *Correlation Id*.

(3) ASP3 sends an *ASPAC* message to SG1 identifying Load Selection LS1 within Application Server AS1/RC1 and receives an acknowledgment and a notification. Data is broadcast the SG and ASP1 and ASP3 for Load Selection LS1 within AS1. The initial Data Messages for LS1 within AS1 are tagged with a *Correlation Id*.

(4) ASP4 sends an *ASPIA* message to SG1 identifying Load Selection LS1 and LS2 within Application Server AS1/RC1 and receives an acknowledgment and a notification.

(5) The same exchange is repeated for SG2.

5.4.2. Failure of ASP1 and ASP2

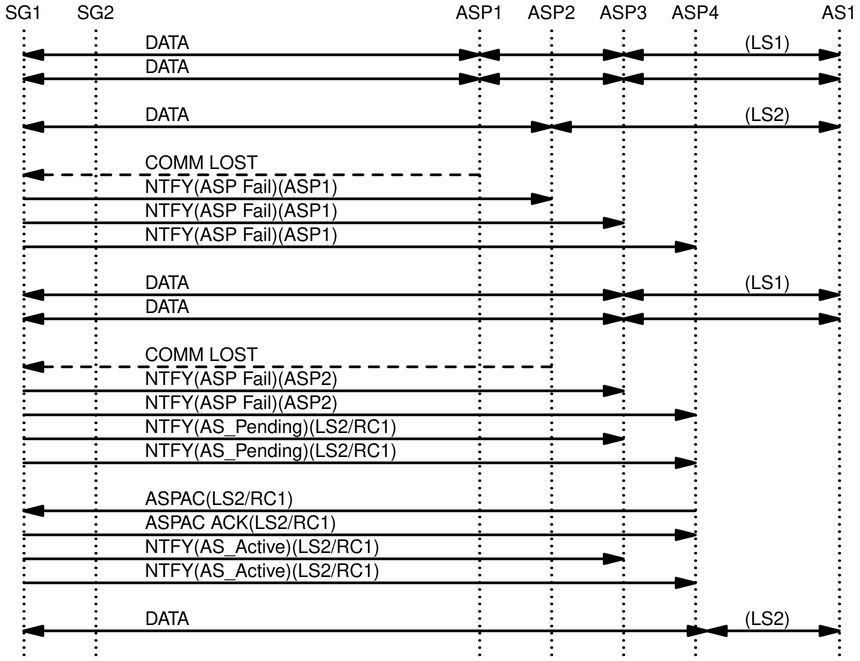

Figure 12. TUA Example – Failure of ASP1

Figure 12 illustrates the failure of ASP1 followed by the failure of ASP2. The sequence of events is as follows:

(1) Data for LS1 within AS1 is broadcast to ASP1 and ASP3. Data for LS2 within AS1 is exchanged with ASP2.

(2) Communication is lost between SG1 and ASP1. ASP2, ASP3, and ASP4 are notified of the failure of ASP1. Data for LS1 in AS1 is directed toward ASP3 only. Data for LS2 in AS1 is unaffected.

(3) Communication is lost between SG1 and ASP2. ASP3 and ASP4 are notified of the failure of ASP1 as well of the AS1 state change to **AS-PENDING**. Data for LS2 is queued at the SG.

(4) ASP4 (spare) responds to the **AS-PENDING** notification and activates for LS2 in AS1/RC1. ASP3 and ASP4 receive an **AS-ACTIVE** notification. Data for LS2 in AS1 is now exchanged with ASP4. Initial DATA messages for LS2 in AS1 are tagged with a *Correlation Id*.

5.4.3. Sparing

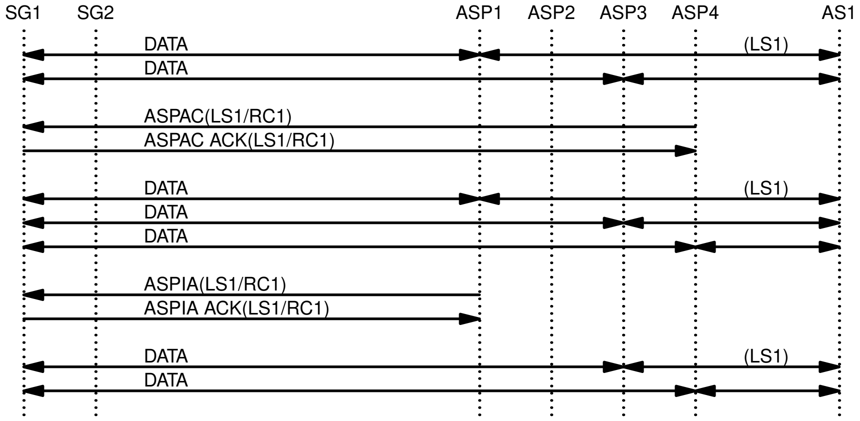

Figure 13. TUA Example – Sparing

Figure 13 illustrates a sparing situation where one ASP takes over traffic from another so that the original ASP can be taken out of service. The sequence of events is as follows:

(1) Data for LS1 in AS1 is broadcast from SG1 to ASP1 and ASP3.

(2) ASP4 (spare) activates for LS1 in AS1 and receives and acknowledgment. Data for LS1 in AS1 is now being broadcast from SG1 to ASP1, ASP3 and ASP4. Initial data for LS1 in AS1 is tagged with a *Correlation Id*.

(3) ASP1 deactivates for LS1 in AS1 and receives and acknowledgment but no notification.

(4) Data or LS1 in AS1 is now broadcast from SG1 to ASP3 and ASP4.

(5) ASP1 can now be taken down and out of service.

6. Security

LOADSEL does not introduce any new security risks or considerations that are not already inherent in the UA [IUA...TUA] Please see the "Security" sections of the UA [IUA...TUA] for security considerations and recommendations that are applicable to each UA.

7. IANA Considerations

LOADSEL adds the following parameter tag value (described in Section 3) to the Common Parameter numbering range of the SIGTRAN UAs [IUA...TUA]:

Tag Value	Parameter Name
0x0018	Load Selector
0x0019	Load Selection

EDITOR'S NOTE:- The *Load Selector* and *Load Selection* tag values shown throughout this document as `0x0018`) and `0x0019`) will be assigned by IANA within the common parameter range of the SIGTRAN UAs and may change its value in further versions of this document.

LOADSEL adds the following value to the *Error Code* parameter (described in Section 3 and 4) for the SIGTRAN UAs [IUA...TUA].

29 Invalid Load Selector

EDITOR'S NOTE:- The *Error Code* value shown throughout this document as `29`) will be assigned by IANA as a value of the common *Error Code* parameter for SIGTRAN UAs and may change its value in further versions of this document.

LOADSEL adds the following values to the *Registration Status* field of the *Registration Result* parameter for the UAs [M2UA...TUA].

12 Error – Unsupported Load Selection
13 Error – Invalid Load Selection
14 Error – Cannot Support Unique Loadsharing
15 Error – Unsupported LS Parameter Field

EDITOR'S NOTE:- The *Registration Status* values shown as `12`, `13`, `14` and `15` above will be assigned by IANA as a value of each UA-specific *Registration Status* parameter for each SIGTRAN UA and may change its value in further versions of this document.

Acknowledgments

The authors would like to thank Tolga Asveren, Ken Morneault, Barry Nagelberg, Benjamin J. Wilson, Jacques Rajchgod, Greg Sidebottom and Gery Verwimp for their valuable comments and suggestions.

Notes

[1] Another draft: *UA Load Grouping* [LOADGRP], provides for selection of *Load Distribution* methods within a *Load Selector*. The draft [LOADGRP] refers to a group of ASPs within the same traffic load selection as a *Load Group* and associates a *Load Distribution* with the load group that can be: Override, Load-share or Broadcast. *LOADSEL* is applicable both to the normal *Traffic Mode Type* of an AS, as well as to the *Load Distribution* within a *Load Group*.

[2] For a detailed description of these messages, see Section 3 of the SIGTRAN UA specifications cited under "References" [IUA...TUA].

[3] For a detailed description of these procedures, see Section 4 of the SIGTRAN UA specifications cited under "References" [IUA...TUA].

[4] **EXAMPLE:-** An ASP (e.g, ASP-1) moving to state **ASP-INACTIVE** within a *Load Selector* (e.g, LS-1) results in the state of the AS moving to **AS-PENDING**. The SGP sends NTFY("Application Server Pending") with "LS-1" in the *Load Selector* parameter in the *NTFY* message. If immediately after (and before T(r) expires) another ASP (e.g, ASP-2) moves to state **ASP-INACTIVE** within another *Load Selector* (e.g, LS-2) resulting the state of the AS being "held" in the **AS-PENDING** state, the SGP sends NTFY("Application Server Pending") again, but this time includes both "LS-1" and "LS-2" in the *Load Selector* parameter in the *NTFY* message.

EXAMPLE:- When all ASPs are inactive for a Load Selection within an Override AS, the AS will transition to the **AS-PENDING state**. The SGP will send a NTFY("Application Server Pending") message to all ASPs configured for the Application Server that are not in the **ASP-DOWN** state. This message should include in the *Load Selector* parameter the Load Selections in which there is no ASP in the **ASP-ACTIVE** state for the AS.

References

IUA. K. Morneault, S. Rengasami, M. Kalla and G. Sidebottom, "ISDN Q.921-User Adaptation Layer," RFC 3057, The Internet Society (November, 2000). [Normative]

DUA. R. Mukundan, N. Mangalpally, K. Morneault and A. Vydyam, "DPNSS/DASS 2 Extensions to the IUA Protocol," <draft-ietf-sigtran-dua-04.txt>, Internet Engineering Task Force - Signalling Transport Working Group (October 2002). Work In Progress. [Informative]

V5UA. E. Weilandt, N. Khanchandani and S. Rao, "V5.2-User Adaption Layer (V5UA)," <draft-ietf-sigtran-v5ua-03.txt>, Internet Engineering Task Force - Signalling Transport Working Group (June 2002). Work In Progress. [Informative]

GR303UA.
R. Mukundan, K. Morneault, "GR-303 extensions to the IUA Protocol," <draft-ranjith-sigtran-gr303ua-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (October 2002). Work In Progress. [Informative]

M2UA. K. Morneault, R. Dantu, G. Sidebottom, B. Bidulock and J. Heitz, “Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) - User Adaptation Layer,” RFC 3331, Internet Engineering Task Force - Signalling Transport Working Group (September, 2002). [Normative]

M3UA. G. Sidebottom, K. Morneault and J. Pastor-Balbas, (eds), “Signaling System 7 (SS7) Message Transfer Part 3 (MTP3) - User Adaptation Layer (M3UA),” RFC 3332, Internet Engineering Task Force - Signalling Transport Working Group (September, 2002). [Normative]

SUA. J. Loughney, G. Sidebottom, L. Coene, G. Verwimp, J. Keller and B. Bidulock, “SS7 SCCP-User Adaptation Layer (SUA),” <draft-ietf-sigtran-sua-14.txt>, Internet Engineering Task Force - Signalling Transport Working Group (June 30, 2002). Work In Progress. [Normative]

ISUA. B. Bidulock, “SS7 ISUP-User Adaptation Layer (ISUA),” <draft-bidulock-sigtran-isua-00.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 5, 2003). Work In Progress. [Informative]

TUA. B. Bidulock, “SS7 TCAP-User Adaptation Layer (TUA),” <draft-bidulock-sigtran-tua-01.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2, 2003). Work In Progress. [Informative]

LOADGRP.
B. Bidulock, “Load Grouping Extension for Signalling User Adaptation Layers (LOADGRP),” <draft-bidulock-sigtran-loadgrp-01.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2, 2003). Work In Progress. [Informative]

RFC 2960.
R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, H. Kalla, L. Zhang and V. Paxson, “Stream Control Transmission Protocol (SCTP),” RFC 2960, The Internet Society (February 2000). [Normative]

Q.704. ITU, “Message Transfer Part – Signalling Network Functions and Messages,” ITU-T Recommendation Q.704, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). [Informative]

Q.723. ITU, “Signalling System No. 7 – Telephone User Part – Formats and codes,” ITU-T Recommendation Q.723, ITU-T Telecommunication Standardization Sector of ITU, Geneva (November 1988). [Informative]

Q.763. ITU, “Signalling System No. 7 – Formats and Codes of the ISDN User Part,” ITU-T Recommendation Q.763, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). [Informative]

Q.713. ITU, “Signalling Connection Control Part Formats and Codes,” ITU-T Recommendation Q.713, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). [Informative]

Q.773. ITU, “Signalling System No. 7 – Transaction Capabilities Formats and Encoding,” ITU-T Recommendation Q.773, ITU-T Telecommunication Standardization Sector of ITU, Geneva (March 1993). [Informative]

RFC 2119.
S. Bradner, “Key words for use in RFCs to Indicate Requirement Levels,” RFC 2119 - BCP 14, Internet Engineering Task Force (March 1997). [Normative]

ASPEXT.
B. Bidulock, “Application Server Process (ASP) Extension Framework,” <draft-bidulock-sigtran-aspext-01.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2, 2003). Work In Progress. [Informative]

LOADSEL.
B. Bidulock, “Load Selection Extension for Signalling User Adaptation Layers (LOADSEL),” <draft-bidulock-sigtran-loadsel-01.txt>, Internet Engineering Task Force - Signalling Transport Working Group (January 2, 2003). Work In Progress. [Normative]

Author's Addresses

Brian Bidulock
OpenSS7 Corporation
1469 Jeffreys Crescent
Edmonton, AB T6L 6T1
Canada

Phone: +1-780-490-1141
Email: bidulock@openss7.org
URL: http://www.openss7.org/

This draft expires July, 2003.

List of Illustrations

Table of Contents

Copyright Statement

Copyright © The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedure for copyrights defined in the Internet Standards process must be followed, or as required to translate into languages other than English.

The limited permission granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and **THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.**